**GITHUB:** <https://github.com/Jose6-sys/cmcs>

**Lecturer Feedback Implementation Report CMCS Project**

I went through each recommendation carefully and made several improvements from the database setup to the unit testing stage. This report explains how I implemented all the changes and fixed the issues that were highlighted.

The first issue I worked on was the database connection. Initially, my application couldn’t connect properly to the database because of a misconfigured connection string. I went back to the appsettings.json file and corrected the connection string to point to the right SQL Server instance. I also made sure the ApplicationDbContext was registered in Program.cs using dependency injection. Once this was done, I was able to store and retrieve data successfully, and the application could manage users and claims without errors.

Next, I focused on improving the models. The feedback mentioned that some of the model properties were missing or incomplete. For example, in the User model, properties like Password and Role were required but not included during testing. I fixed this by adding those properties and marking them as required. I also made sure that the Claim model had important properties such as Notes, HoursWorked, and HourlyRate properly set up. This helped ensure that the database had all the necessary information for each record and that validation worked correctly.

After that, I moved on to the controllers, especially the ClaimsController. I received feedback that some of the controller methods were not handling certain inputs properly. In the SubmitClaim method, for example, there was a problem with file uploads and argument types. I corrected this by making sure the parameter for the uploaded file used the correct type IFormFile instead of a string. I also added proper null checks and validations to prevent errors if no file was uploaded. For the UpdateStatus method, I made sure the logic correctly updated the claim’s status in the database and handled missing data properly.

Once I had the database and controllers working, I moved on to unit testing. At first, the tests failed because there was an ambiguous reference between cmcs.Models.Claim and System.Security.Claims.Claim. To fix that, I used a namespace alias using ClaimModel = cmcs.Models.Claim so that the tests would point to the correct model.

I built both the main project and the test project again in Visual Studio. Initially, I had to add a project reference from the test project to the main cmcs project to make sure all classes were recognized. Once that was done, the build succeeded. I then ran the tests using the command dotnet test in the terminal, and all tests executed as expected.

Through these steps, I was able to implement all the recommendations. My project now has a working database connection, properly structured models, improved controller logic, and fully functional unit tests. This process helped me understand how to properly handle Entity Framework configurations, manage dependencies, and use unit testing to confirm that the system works correctly. It also improved my confidence in debugging and working with real-world .NET MVC applications.